**Caching** :

A Cache is an area of local memory that holds copy of frequently accessed data that could be time-consuming or expensive to compute. It is designed to hold objects in memory for immediate access by application.

It is different from an application accessing the External Storage solutions as the applications have to obtain connection to the database, then execute the query across the network & parse the results returned.

**Features:**

* The items (key-value pairs) will expire based on the time-to-Live period.
* Cache will keep most recently used items if you will try to add more items than the max specified size because of the functionality of LRUMap- removes least used entries from the fixed sized map.
* For the expiration of items, we can timestamp the last access and in a separate thread remove the items when the time to live limit is reached. This is nice for reducing memory pressure for applications that have long idle time in between accessing the cached objects.

**Design:**

**JAVA Class : MemoryCache.java**

Created java function to **define** how to add/delete the key value pairs,

It also calculates least accessed values & removes them automatically after the time to live is reached.

Also, keeps track/record of the most recently used key-value pairs.

**JAVA Class : InMemoryCacheTest.java**

In this function, we are calling the functions defined in JAVA Class : MemoryCache.java,

to add/delete the key value pairs by passing the values dynamically.

It also tests by adding the values after max size limit is reached and retrieving the time interval &

Also cleanup the memory cache for the objects which are older than the time-to-live which is being passed.

**Trade-Offs:**

* The program depicts the basic functionality of the Caching
* Can be integrated with Web Applications for the WebCenter Sites Caching Management & improve response.